# Thomas Jones – CS5530-0002 – Semester Project

## Overview

The project consisted of the analysis of 3 datasets; laptop price prediction using regression, student success prediction using classification, and analysis of apartment distributions via clustering. This document outlines the data collection, pre-processing, data analysis, model development, performance evaluation, and interpretation and conclusions for each dataset.

## Regression Discussion

The purpose of the linear regression exercise is to accurately predict the price of a laptop based on the features.

### Data Collection

The data itself was provided as part of the project. The CSV was stored in github and loaded from there.

### Pre-Processing

The first step in pre-processing is to review the data. We’ll take a look at the data types and missing values for each column.

Data Types

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Company 1275 non-null object

1 TypeName 1275 non-null object

2 Inches 1275 non-null float64

3 Ram 1275 non-null int64

4 OS 1275 non-null object

5 Weight 1275 non-null float64

6 Price\_euros 1271 non-null float64

7 Screen 1275 non-null object

8 ScreenW 1275 non-null int64

9 ScreenH 1275 non-null int64

10 Touchscreen 1275 non-null object

11 IPSpanel 1275 non-null object

12 RetinaDisplay 1273 non-null object

13 CPU\_company 1275 non-null object

14 CPU\_freq 1274 non-null float64

15 CPU\_model 1275 non-null object

16 PrimaryStorage 1275 non-null int64

17 SecondaryStorage 1275 non-null int64

18 PrimaryStorageType 1275 non-null object

19 SecondaryStorageType 1275 non-null object

20 GPU\_company 1275 non-null object

21 GPU\_model 1275 non-null object

Null Values

Company 0

TypeName 0

Inches 0

Ram 0

OS 0

Weight 0

Price\_euros 4

Screen 0

ScreenW 0

ScreenH 0

Touchscreen 0

IPSpanel 0

RetinaDisplay 2

CPU\_company 0

CPU\_freq 1

CPU\_model 0

PrimaryStorage 0

SecondaryStorage 0

PrimaryStorageType 0

SecondaryStorageType 0

GPU\_company 0

GPU\_model 0

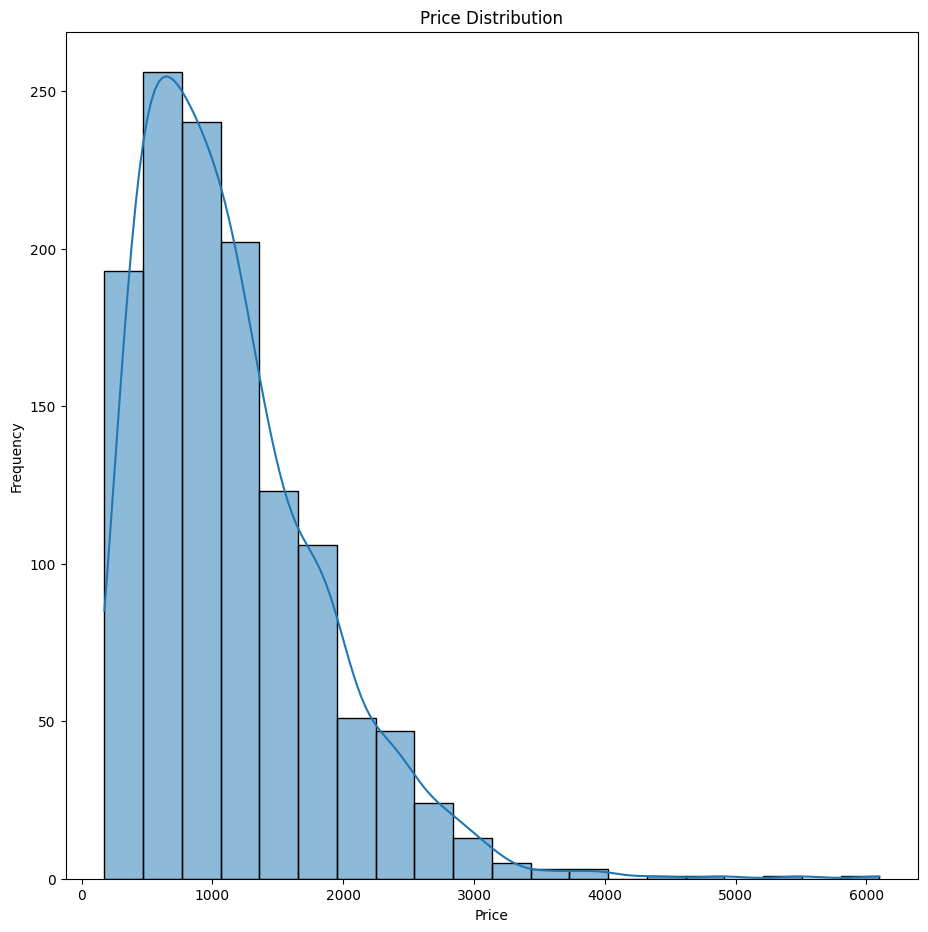
Key Observations

* Since price is our target variable it does not make sense to impute those. Those entries should be dropped.
* For RetinaDisplay, it is reasonable to assume that missing data means "no" so for those values we will use default values.
* There is only a single value missing for CPU Frequency, as this is not significant compared to the total data size we will remove that row.

### Analysis

#### Price Analysis

Let's first take a look at the prices. This will give us an idea of how many laptops we'll find at each price threshold.



Due to the low counts of data at the higher prices we can expect that the regression models will favor accurate predictions for prices up to around 4000 Euros but past that the price prediction will likely be inaccurate. In a production environment we would likely model the data across a variety of price ranges, possibly choosing different models based on the number of samples at the higher range. For purposes of this exercise, there are a couple of choices we can make, we can either limit the model to laptops of 4000 Euros or less in order to provide more accurate predictions or we can attempt to model the entire dataset but be prepared for worse results as price increases. For this exercise we will limit the price to 4000 Euros.

Next, we find that the laptop data contains a mix of category and numerical data. We'll start with the category data first to get an idea of what values are present for each category. To do this, we'll first split out the category columns and then view the unique values for each category.

Unique Counts

Company 19

TypeName 6

OS 9

Screen 4

Touchscreen 3

IPSpanel 2

RetinaDisplay 2

CPU\_company 3

CPU\_model 91

PrimaryStorageType 4

SecondaryStorageType 4

GPU\_company 4

GPU\_model 110

#### CPU/GPU Analysis

One thing to notice is the large differences in counts of different types of CPUs and GPUs. These may be useful for predictive analysis later but that data won't fit very well on bar charts. It may be that there are a few types that dominate the counts, however, so we can evaluate them.

A comparison of a graph

Description automatically generated

For the plot above we can see that there are a large number of CPU and GPU models that exist on the tails of the plots, i.e. there are a large number that have only a few values. We can group those together into an "other" category. This will allow us to utilize the model as a predictive characteristic for the more common cases.

We then get the following distribution,

A comparison of a graph

Description automatically generated

Unique Counts

Company 19

TypeName 6

OS 9

Screen 4

Touchscreen 3

IPSpanel 2

RetinaDisplay 2

CPU\_company 3

CPU\_model 91

PrimaryStorageType 4

SecondaryStorageType 4

GPU\_company 4

GPU\_model 110

CPU\_model\_adj 15

GPU\_model\_adj 16

And the price distributions for the GPU and CPU models.

A graph of blue and white bars

Description automatically generated

A graph of blue and white boxes

Description automatically generated with medium confidence

The above graphs show us that there is some relationship between price and GPU model. Our "Other" category does show a few outliers but since these are singular values (other than the cluster around 4000), it's unlikely that there will be an enormous effect on the regression model given the other features.

#### Remaining Categories

We can also visualize the other categorical data across different categories. We can split it out by company.

A graph of blue and white bars

Description automatically generated with medium confidence

And the remaining categories.

A group of blue and white boxes with text

Description automatically generated

From the box plots above we can see clear trends in price for some of the categories.

#### Numerical Analysis

Next we're going to look at the numerical data for the dataset.

A group of graphs showing different sizes of data

Description automatically generated with medium confidence

We can clearly see some outliers present in Weight, and Ram that are likely affecting the distribution. We can remove those by using the Z score and for this exercise we’ll use a Z score limit of 5.

Removing the outliers then gives us the following:

A group of blue dots

Description automatically generated

We can see now as well that we have a better distribution for weight and while it’s not obviously linear (or even poly) there is a visual trend.

The final step is to scale the numeric data using the StandardScaler. Standard scaling was chosen as none of the features fall within a fixed range, i.e. percentages.

### Modeling and Results

#### Linear Modeling

The first model chose was linear regression. Linear regression is best suited to numeric data so the categorical data was not included in the datapoints. We first run a linear regression against the numeric data, holding out 20% for testing which yields the following regression and test results.

Intercept: 1125.785860694618

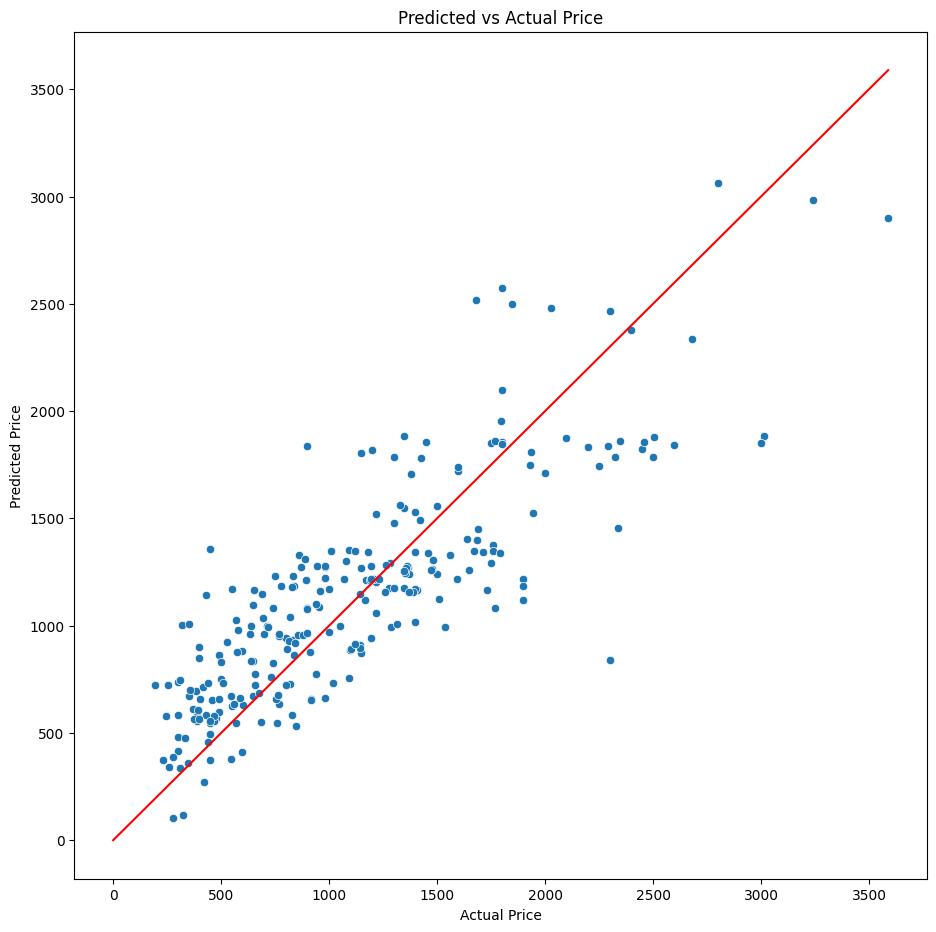
Coefficients: [-107.44016466 351.62095825 45.52864488 -39.56923085 217.65087398

146.82384665 -71.16953571 -8.98879236]

MSE: 125443.41062094783

R^2: 0.6928408559710251

The points, residuals and the histogram of the residuals are also plotted.



A graph with blue dots

Description automatically generated

A graph of a number of blue bars

Description automatically generated

The residual mean and standard deviation were calculated yielding,

Residual Mean: -27.910561491311622

Residual Std: 353.7783362821209

So our model shows a bias of -27 Euros against the test data with the other 68% of the test values being within 353 Euros of the actual price. The score in this case was somewhat low with R2 only being 69%. Looking at the spread of the error it should be obvious that as the price increases the error increases.

We can evaluate other ways to expand the features as well. For example, if we do a polyfit with degree 2 on the numeric data we get the following test statistics.

Intercept: 1087.993873551933

Coefficients: [-3.33189235e-11 9.94467723e+00 3.25328267e+02 -1.44311977e+02

-3.01285601e+02 4.70191733e+02 1.84160936e+02 3.69665031e+01

4.66483885e+01 4.23755508e+01 -2.23245112e+01 -6.40141300e+01

2.35807969e+01 7.04499151e+01 -9.11965365e+01 1.37353961e+01

1.07100396e+02 -5.10654677e+01 1.86004643e-01 -6.75498142e+01

8.11960416e+01 6.81400708e+01 5.78305960e+00 5.40462342e+00

1.70396103e+02 -4.72335925e+02 4.63198455e+02 9.00700200e+01

-9.12060511e+01 -1.47348143e+02 9.22603487e+02 -1.77415794e+03

-4.42689432e+01 -5.09635789e+02 1.39957163e+02 8.51700559e+02

-7.59849447e+00 5.34690679e+02 -1.87730145e+02 1.99361028e+01

-4.22593989e+01 1.11816364e+00 -2.15611323e+01 7.45608547e+01

-6.12523211e+00]

MSE: 90386.51733813663

R^2: 0.7786807201756201

We can see that the score of the fit has significantly increased.

A graph with blue dots and red line

Description automatically generated

A graph with blue dots

Description automatically generated

A graph of a distribution of a number of individuals

Description automatically generated with medium confidence

The mean and standard deviation of the residuals is now,

Residual Mean: -41.20425922331162

Residual Std: 298.3968259722482

The bias increased for the model, but the overall variance of the error has decreased as well. It is likely that there some non-linearity in the numeric data we are not quite capturing right. However, we are not including any of the categorical features which likely have just as a significant of impact.

We can include categorization in a linear regression using one-hot encoding, however, initial experimentation (ad-hoc, not included in this document) showed either failed regressions or poor results. We can use PCA though to smoot out the individual effects of the one hot-encoded data. The first step in this process was to one-hot encode the categories. Next, the numeric data was expanded with a poly fit of degree 2 and the full data combined. This yields the following shapes:

Before shape: (1263, 24)

After dropping CPU and GPU model columns shape: (1263, 22)

After dropping numeric columns shape: (1263, 13)

One Hot Encoded + Categories shape: (1263, 91)

Before Numeric shape: (1263, 9)

After dropping Price shape: (1263, 8)

Shape of polyfit numeric data: (1263, 45)

Total shape: (1263, 136)

Again, through experimentation not included in this doc, it was found that a linear model based on the top 109 PCA components. At 110 the regression fails, though this might be a library or environment issue.

A graph of a function

Description automatically generated

Because regression is a fast operation we can use all 109 components in the analysis. This yields the following test results (intercept and coefficients available in notebook).

MSE: 56934.699301350505

R^2: 0.8605904174927657

The score has again increased significantly with the addition of the categorical data.

A graph with blue dots

Description automatically generated

A graph with blue dots

Description automatically generated

A graph of a distribution of a number of individuals

Description automatically generated with medium confidence

The new residual mean and standard deviation are:

Residual Mean: 1.0228429751484642

Residual Std: 239.08069827657474

Our bias is now almost eliminated and the overall errors are within 240 Euros for our estimates.

#### Decision Tree Modeling

We next modeled the data with a decision tree. Decision trees are better able to handle categorical data, we can use the full dataset. To keep things simpler we'll use 100 estimators, the default.

This produced the following metrics:

MSE: 45688.127641567204

R^2: 0.8881286301990352

Residual Mean: -25.893591887822307

Residual Std: 212.59419325829552

The errors are biased low but the standard deviation continues to reduce vs the linear models.

A graph with blue dots

Description automatically generated

A graph with blue dots and a red line

Description automatically generated

A graph of a distribution of a number of individuals

Description automatically generated with medium confidence

We continue to find the characteristic spread of error as price increases.

#### MLP Modeling

The final model was an MLP. We chose 2 hidden layers and iterated from 10 to 400 units per layer to find the best fit for layer size, which for this data was 260.

A graph with blue lines

Description automatically generated

This architecture product the following metrics:

MSE: 43888.356310199844

R^2: 0.8925355274513838

Residual Mean: -0.8925582107946364

Residual Std: 209.9088293349721

The bias is now nearly gone and the standard deviation of the error the lowest of the models.

A graph with blue dots and red line

Description automatically generated A graph with blue dots

Description automatically generated

A graph of a distribution of a number

Description automatically generated with medium confidence

### Conclusion

The higher complexity models (Decision Tree and MLP) produced better results for the given data. This falls in line with general industry observations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | MSE | R2 | Residual Mean | Residual Std |
| Linear | 125443 | 0.623 | -27.91 | 335.78 |
| Linear-Poly | 90386 | 0.778 | -41.20 | 298.40 |
| Linear-PCA | 56934 | 0.861 | 1.02 | 239.08 |
| Decision Tree | 45425 | 0.889 | -25.70 | 212.00 |
| MLP | 43888 | 0.893 | -0.89 | 209.91 |

## Classification Discussion

The classification dataset consists of student data; personal, performant, and environmental with a target column indicating if the student graduated, dropped out, or enrolled. For this exercise we want to develop a model that provides a prediction if the student will drop out or graduate.

### Data Collection

The data itself was provided as part of the project. The CSV was stored in github and loaded from there.

### Pre-Processing and Analysis

First, we’ll take a look at the data types available.

Data Types

0 Marital status 4424 non-null int64

1 Application mode 4424 non-null int64

2 Application order 4423 non-null float64

3 Course 4422 non-null float64

4 Daytime/evening attendance 4421 non-null float64

5 Previous qualification 4421 non-null float64

6 Previous qualification (grade) 4419 non-null float64

7 Nacionality 4419 non-null float64

8 Mother's qualification 4420 non-null float64

9 Father's qualification 4420 non-null float64

10 Mother's occupation 4419 non-null float64

11 Father's occupation 4420 non-null float64

12 Admission grade 4421 non-null float64

13 Displaced 4421 non-null float64

14 Educational special needs 4421 non-null float64

15 Debtor 4421 non-null float64

16 Tuition fees up to date 4422 non-null float64

17 Gender 4422 non-null float64

18 Scholarship holder 4422 non-null float64

19 Age at enrollment 4424 non-null int64

20 International 4424 non-null int64

21 Curricular units 1st sem (credited) 4424 non-null int64

22 Curricular units 1st sem (enrolled) 4424 non-null int64

23 Curricular units 1st sem (evaluations) 4424 non-null int64

24 Curricular units 1st sem (approved) 4424 non-null int64

25 Curricular units 1st sem (grade) 4424 non-null float64

26 Curricular units 1st sem (without evaluations) 4424 non-null int64

27 Curricular units 2nd sem (credited) 4424 non-null int64

28 Curricular units 2nd sem (enrolled) 4424 non-null int64

29 Curricular units 2nd sem (evaluations) 4424 non-null int64

30 Curricular units 2nd sem (approved) 4424 non-null int64

31 Curricular units 2nd sem (grade) 4424 non-null float64

32 Curricular units 2nd sem (without evaluations) 4424 non-null int64

33 Unemployment rate 4424 non-null float64

34 Inflation rate 4424 non-null float64

35 GDP 4424 non-null float64

36 Target 4419 non-null object

And the null counts

Null Counts

Marital status 0

Application mode 0

Application order 1

Course 2

Daytime/evening attendance\t 3

Previous qualification 3

Previous qualification (grade) 5

Nacionality 5

Mother's qualification 4

Father's qualification 4

Mother's occupation 5

Father's occupation 4

Admission grade 3

Displaced 3

Educational special needs 3

Debtor 3

Tuition fees up to date 2

Gender 2

Scholarship holder 2

Age at enrollment 0

International 0

Curricular units 1st sem (credited) 0

Curricular units 1st sem (enrolled) 0

Curricular units 1st sem (evaluations) 0

Curricular units 1st sem (approved) 0

Curricular units 1st sem (grade) 0

Curricular units 1st sem (without evaluations) 0

Curricular units 2nd sem (credited) 0

Curricular units 2nd sem (enrolled) 0

Curricular units 2nd sem (evaluations) 0

Curricular units 2nd sem (approved) 0

Curricular units 2nd sem (grade) 0

Curricular units 2nd sem (without evaluations) 0

Unemployment rate 0

Inflation rate 0

GDP 0

Target 5

We can see from the numbers above that for the given 4000+ samples we have very few missing values. Since that is the case we can simply drop those rows with missing data. We also don’t need the rows where the student is Enrolled. We are trying to develop a model to predict if the student will Graduate or Dropout so Enrolled does not contribute to our model.

Now we have a little over 3600 rows remaining with only Dropout and Graduate students and no missing values.

We can also review the unique values for each column to better understand the data.

Unique Values

Marital status 6

Application mode 18

Application order 8

Course 17

Daytime/evening attendance\t 2

Previous qualification 17

Previous qualification (grade) 101

Nacionality 21

Mother's qualification 29

Father's qualification 34

Mother's occupation 33

Father's occupation 46

Admission grade 619

Displaced 2

Educational special needs 3

Debtor 2

Tuition fees up to date 2

Gender 2

Scholarship holder 2

Age at enrollment 46

International 2

Curricular units 1st sem (credited) 21

Curricular units 1st sem (enrolled) 23

Curricular units 1st sem (evaluations) 35

Curricular units 1st sem (approved) 23

Curricular units 1st sem (grade) 797

Curricular units 1st sem (without evaluations) 11

Curricular units 2nd sem (credited) 19

Curricular units 2nd sem (enrolled) 23

Curricular units 2nd sem (evaluations) 30

Curricular units 2nd sem (approved) 20

Curricular units 2nd sem (grade) 782

Curricular units 2nd sem (without evaluations) 10

Unemployment rate 11

Inflation rate 9

GDP 10

Target 3

Looking at the data it appears that most of the data is likely category data that has been encoded already. For example, Marital Status.

Marital Status

1 3919

2 379

4 91

5 25

6 6

Listing course as well:

Course

9500.0 765

9147.0 379

9238.0 355

9085.0 337

9773.0 331

9670.0 268

9991.0 268

9254.0 252

9070.0 226

171.0 215

8014.0 215

9003.0 210

9853.0 192

9119.0 170

9130.0 141

9556.0 86

33.0 12

We find that while this is a numeric column in datatype, it is really categorical in nature. We'll deal with this later before we start developing the models.

Let's plot the data against Target. Since there are so many columns we'll break it into separate plots. We have 37 different columns, but the last column is Target so that gives us 36. We can do 3 3x4 plots.

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated

There are some columns that we can drop as they are heavily asymmetric with regards to sample size.

* Marital status
* Nacionality
* Educational special needs
* International
* Curricular units 1st sem (credited)
* Curricular units 2nd sem (credited)

There are also two columns that have ethical consideration as well, namely

* Gender
* Age at enrollment

We'll drop the first set of columns and produce two models for each of the second.

Also, as we saw earlier, the course column will be problematic if we simply scale it as coded. To get around this we'll re-code the values 1-17 (the number of unique courses).

### Modeling & Results

For all models the test set was split to be 20% of the total data.

#### Logistic Regression

A simple logistic regression model was created. The confusion matrix and classification report for both the asymmetric and ethical data was.

**Asymmetric:**

Validation Data Accuracy: 0.8977900552486188

Confusion Matrix:

[[236 54]

[ 20 414]]

Classification Report:

precision recall f1-score support

Dropout 0.92 0.81 0.86 290

Graduate 0.88 0.95 0.92 434

accuracy 0.90 724

macro avg 0.90 0.88 0.89 724

weighted avg 0.90 0.90 0.90 724

**Ethical:**

Validation Data Accuracy: 0.893646408839779

Confusion Matrix:

[[233 57]

[ 20 414]]

Classification Report:

precision recall f1-score support

Dropout 0.92 0.80 0.86 290

Graduate 0.88 0.95 0.91 434

accuracy 0.89 724

macro avg 0.90 0.88 0.89 724

weighted avg 0.90 0.89 0.89 724

We can see from the results that there is very little difference in model quality when we exclude some of the PII data such as gender. This tells us that for a production prediction we can drop these columns, and probably should.

#### Support Vector Classifier

A support vector classifier model was developed and the value of C selected by first using k-fold validation over a range of values from 0.1 to 5 in steps of 0.1. The C producing the best value for the average best accuracy of the folds for each value of C was selected.

A graph of a graph

Description automatically generated

The best value for C was found to be 2 for this analysis.

Best Accuracy: 0.906429634923471

**Asymmetric:**

Validation Data Accuracy: 0.9019337016574586

Confusion Matrix:

[[233 57]

[ 14 420]]

Classification Report:

precision recall f1-score support

Dropout 0.94 0.80 0.87 290

Graduate 0.88 0.97 0.92 434

accuracy 0.90 724

macro avg 0.91 0.89 0.89 724

weighted avg 0.91 0.90 0.90 724

**Ethical:**

Validation Data Accuracy: 0.9005524861878453

Confusion Matrix:

[[229 61]

[ 11 423]]

Classification Report:

precision recall f1-score support

Dropout 0.95 0.79 0.86 290

...

accuracy 0.90 724

macro avg 0.91 0.88 0.89 724

weighted avg 0.91 0.90 0.90 724

We achieved a slightly better result from the SVC than we did for the logistic regression. Depending on the accuracy expectations and the use of the model we might choose the simpler model. For example, if the model were being used to target certain students for additional advisement or monitoring, i.e. "at risk" students, the lower accuracy might be acceptable. However, if the model were used as a tool for acceptance we would want to significantly increase the accuracy.

#### K Nearest Neighbors

For the KNN models a different approach to selecting the best K was done. Instead of K fold, the model was run over different values of K from 2 to 30. This was done for each of the datasets, asymmetric and ethical, and the K providing the best accuracy was selected for each.

**Asymmetric (K=16):**

Validation Data Accuracy: 0.8549723756906077

Confusion Matrix:

[[200 90]

[ 15 419]]

Classification Report:

precision recall f1-score support

Dropout 0.93 0.69 0.79 290

Graduate 0.82 0.97 0.89 434

accuracy 0.85 724

macro avg 0.88 0.83 0.84 724

weighted avg 0.87 0.85 0.85 724

A graph of a number of neighbors

Description automatically generated

**Ethical (K=8):**

Validation Data Accuracy: 0.8715469613259669

Confusion Matrix:

[[215 75]

[ 18 416]]

Classification Report:

precision recall f1-score support

Dropout 0.92 0.74 0.82 290

Graduate 0.85 0.96 0.90 434

accuracy 0.87 724

macro avg 0.88 0.85 0.86 724

weighted avg 0.88 0.87 0.87 724

A graph of a number of neighbors

Description automatically generated

The KNN model shows a noticeable decrease in performance compared to the logistic and SVC models.

#### Additional Classifiers

Random Forest and MLP classifiers were also tested against the dataset. The results for these can be found in the notebook.

### Interpretation

We can use PCA to evaluate the data to help understand the separability of the two classes. As can be seen in the plots, the PCA plots visually show the some of the difficulty of getting higher than 90% accuracy with the models. While there are clear clusters of separation in the data, i.e. there are obvious cases where a given student will likely succeed or not, the overlap shows the complexity of the underlying source of the data, i.e. people. The models are not absolute predictors and no model will ultimately be. At best they are estimators and that fact must be taken into consideration. For this analysis a component count of 20 was selected of the 36 available features. The plots were done for both the asymmetrical and ethical data.

#### Asymmetric

A graph with a line

Description automatically generated

A diagram of a number of dots

Description automatically generated with medium confidence

#### Ethical

A graph with a line

Description automatically generated

A diagram of a number of dots

Description automatically generated with medium confidence

### Conclusion

The ROC curves for each of the models show that for the models used, there was little difference in model capabilities except for KNN which under-performed the other 4 models.

A graph of a curve

Description automatically generated

A graph of a curve

Description automatically generated with medium confidence

## Clustering Discussion

The clustering exercise consists of apartment data across the United States. We will use clustering to identify any patterns that might exist within the data.

### Data Collection

The data itself was provided as part of the project. The CSV was stored in github and loaded from there.

### Pre-Processing

The data consists of 10 columns of data about the apartments.

Data Types

0 id 99491 non-null float64

1 bathrooms 99427 non-null float64

2 bedrooms 99365 non-null float64

3 fee 99490 non-null object

4 pets\_allowed 39066 non-null object

5 price 99491 non-null float64

6 price\_type 99489 non-null object

7 square\_feet 99491 non-null float64

8 state 99189 non-null object

9 latitude 99465 non-null float64

10 longitude 99466 non-null float64

Null Value Counts

id 1

bathrooms 65

bedrooms 127

fee 2

pets\_allowed 60426

price 1

price\_type 3

square\_feet 1

state 303

latitude 27

longitude 26

The large number of pets\_allowed NaN values tells us we should not likely drop this data. Instead we will assume that a proper value is either "No" or "Not Allowed". First though, we'll see what unique values are in that column.

pets\_allowed

Cats,Dogs 37095

Cats 1843

Dogs 127

Cats,Dogs,None 1

Instead of keeping the data exactly as it is, we'll split the data into dogs allowed and cats allowed columns.

This leaves 99,004 rows. We’ll also look at the unique values for price\_type and fee.

price\_type

Monthly 99001

Weekly 3

fee

No 98807

Yes 197

Price type "Weekly", since there are only 3 rows, is unlikely to contribute to any meaningful analysis so we'll drop the column as that would only leave a single category of data. Same with fee since "Yes" accounts for only about 2% of the values.

### Analysis

Since we have a limited number of columns we can review the pair-plots of the numeric columns.

A graph of data in a row

Description automatically generated with medium confidence

If we look carefully at the latitude vs. longitude data it is (as we would expect after some thought) a map of the United States, including outliers for Alaska and Hawaii. The state and lat/long columns are highly correlated so we can pick one or the other. Since the state is a category and it is possible there are regional variances that we might want modeled we'll drop the state column.

There are also some outliers in bathrooms and bedrooms that we should address. We will do this by dropping any rows with a Z score over 5. Doing this gives us an updated pair plot.

A graph of data on a white background

Description automatically generated

This shows a better distribution of the overall data. We can also see some clear trends in the density data with defined peaks. The final step is to scale the numeric data so that the distance functions are not dominated by one feature over others, for example by price.

### Modeling

#### KMeans Clustering

The first step is to capture the inertia across different values of K.

A graph with a line

Description automatically generated

While there is not a clear "elbow" in the graph, about 10 at least seems reasonable. More than 10 clusters might be overwhelming to show visually as well.

We'll plot the clusters visually using Latitude and Longitude.

A map of different colored dots

Description automatically generated

Because we are including latitude and longitude within the features we do expect some level of localization of clusters by region. We can plot each cluster separately as well. As will be seen in the clusters, the regionality is clearly distinguished but not universal.

A group of colored dots

Description automatically generated

We can look at the centroids for each cluster as well, this will give us a better idea as to what each cluster is really representing.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| bathrooms | bedrooms | price | Square feet | latitude | longitude | Dogs Allowed | Cats Allowed |
| 1.93 | 2.257 | 1599.37 | 1161 | 39.3348 | -82.4211 | 0.355 | 0.369 |
| 1.00 | 1.171 | 1035.61 | 738 | 32.2143 | -91.2768 | 0 | 0.004 |
| 1.02 | 1.329 | 1320.19 | 765 | 39.6068 | -79.7871 | 0.958 | 0.999 |
| 1.10 | 1.405 | 1288.16 | 766 | 42.1094 | -110.6858 | 0.327 | 0.366 |
| 1.01 | 1.371 | 1378.75 | 783 | 39.9820 | -77.9752 | 0 | 0.021 |
| 1.01 | 1.207 | 2095.26 | 717 | 34.3892 | -117.9510 | 0.292 | 0.322 |
| 1.00 | 1.155 | 1036.80 | 732 | 32.5168 | -94.2177 | 0.994 | 0.999 |
| 2.00 | 2.222 | 1353.01 | 1139 | 32.4572 | -94.4049 | 0.436 | 0.439 |
| 2.56 | 3.669 | 2282.48 | 2265 | 36.6096 | -91.5063 | 0.298 | 0.288 |
| 2.00 | 2.230 | 3110.30 | 1203 | 36.9931 | -106.7384 | 0.310 | 0.323 |

Looking at the means we can see, for example, that bathrooms clearly form two major groups around 1 or 2 bathrooms. This general pattern is seen in the other features as well where the means form obvious steps or group within the clusters. Naively we might expect the centroids to represent different geographic regions since we have include latitude and longitude but as seen in the scatter plots, for example clusters 6 and 8, there are some clusters that appear geographically independent.

#### Hierarchical Clustering

For the hierarchical clustering a sample size of 2000, or about 2%, of the data was selected. This number was selected so that the clusters could be visually within the maps.

A graph of colorful dots

Description automatically generated with medium confidence

Then we can plot each cluster separately as well.

A group of colored dots

Description automatically generated

As with the K-Means clustering we can see that many of the clusters are regionally based but there are other clusters present that indicate national trends.

Finally, we present the dendrogram of the clusters.

A diagram of a graph

Description automatically generated with medium confidence

The dendrogram shows that the clusters are generally well spread for the samples taken from the greater population.

## Project Conclusion

The biggest learning I had from this project is the complexity involved in selecting a model. While the linear regression results showed an increased in accuracy as the model complexity increased the clustering model, for example, showed relatively limited or no improvement from the simplest model, logistic regression.

The project also showed the clear role that visual representation of data and play. While accuracy or centroid information is useful, showing the clustering as a map, for example, gives a real-world feel to the data that simple numerics would not accomplish.

Finally, data analysis is fundamental to both of these learnings. The domain of the data and the objective of the model must clearly be understood. Otherwise the results and analysis will be misunderstood, or worse, be mis-used as would be possible with the student classification data using data such as gender as a predictive feature, ethics matter.